

## ON A PROPERTY OF THE NUMBER 977731833235239280

WITOLD JARNICKI AND MACIEJ ŻENCZYKOWSKI

ABSTRACT. We solve a theoretical arithmetics problem stated by Waclaw Sierpiński. The problem has remained open for a couple of decades.

## 1. INTRODUCTION

Consider the following puzzle from number theory, presented almost 50 years ago.

*Problem 1* (W. Sierpiński). Find a composite number such that it remains composite after altering any two digits in its decimal representation.

We found the problem in [Mat 1977]. It was stated as a puzzle for the readers. It turned out (see [Mat 1978]) that no one had solved it.

A ternary version of the problem was stated in [Mat 1978] with a solution given in [Mat 1980]. The basic idea was that altering any two digits in  $40 = 1111_3$  keeps it even and different from 2. Observe that such an assumption (changing exactly two digits) makes the problem easy to solve (see Table 1). Therefore, it is reasonable to assume the following statement.

*Problem 2* (W. Sierpiński). Find a composite number such that it remains composite after altering at most two digits in its base  $b$  representation.

base	allowed		not allowed	
	solution	decimal	solution	decimal
2	1010100	84	1001	9
3	1111	40	11	4
4	20130000	34560	12321230	28268
$> 4$	4	4	4	4

TABLE 1. Minimal solutions to Problem 1, depending on whether one is allowed to change the most significant digit to zero or not.

---

1991 *Mathematics Subject Classification.* 11Y11, 65Y05.

*Key words and phrases.* prime number, composite number, sieve of Atkin, numerical methods.

Part of the computation was done on High Performance Computers of The Academic Computer Centre CYFRONET AGH: SGI Altix 3700 (baribal, grant number MNiSW/SGI3700/UJ/097/2007), IBM BladeCenter HS21 (mars, grant number MNiSW/IBM\_BC\_HS21/UJ/097/2007), and PC Rack-Saver cluster (zeus, grant number MNiSW/PC\_RS/UJ/097/2007).

The authors obtained a significant increase of computing speed by using their program to test newly installed Google machines.

Another question one may ask is whether it is permitted to change the most significant digit to zero. However, disallowing that gives little help in finding the number (see Tables 2 and 3), so it is safer to assume it may be done.

The problem was investigated from the theoretical side in [Sch 1992]. There the author shows, that there are infinitely many solutions to Problem 2 (for any  $b$ ), provided that Erdős’s “favorite” conjecture on covering systems of congruences (see [Erd 1952]) is true. However, since Erdős’s conjecture is open, so remained the problem.

## 2. MAIN RESULTS

We present the solution to Problem 2 (bases 2–10) in Tables 2 and 3.

base	solution	decimal
2	1010100	84
3	2200100	1953
4	20130000	34560
5	3243003420	7000485
6	55111253530	354748446
7	5411665056000	77478704205
8	33254100107730	1878528135128
9	210324811482600	48398467146642
10	977731833235239280	977731833235239280

TABLE 2. Solutions to Problem 2 found for bases 2–10 (changing the most significant digit to zero is allowed). They are known to be minimal for bases 2–9.

base	solution	decimal
2	1010100	84
3	2200100	1953
4	12321230	28268
5	324322330	1401590
6	43040303150	273241578
7	5411665056000	77478704205
8	33254100107730	1878528135128
9	210324811482600	48398467146642
10	977731833235239280	977731833235239280

TABLE 3. Solutions to Problem 2 found for bases 2–10 (changing the most significant digit to zero is disallowed). They are known to be minimal for bases 2–9.

## 3. MOTIVATION AND METHODS

Since there are many ways to modify a number by altering two of its digits, it initially seems impossible to find a solution to the problem. Therefore, a natural way to approach the puzzle is to consider it in bases smaller than 10 — in such situation the number of ways of changing a number is much smaller.

We managed to solve the problem for bases between 2 and 9 using a small “grid” of computers at our university. The computation lasted several weeks. Looking at the results gives grounds to suppose that solving the problem for base  $k+1$  requires about 100 times the computing power needed to solve it for base  $k$ . This estimate inspired us to use grid computing to solve the decimal case.

To present the main idea behind the solution method, consider the following lemma.

**Lemma 3.** *Let  $n$  contain at least 5 nonzero digits in base  $b$ . Assume that  $n$  is a solution to Problem 2. Define  $\tilde{n} := b \lfloor n/b \rfloor$ . Then  $\tilde{n}$  is a solution to Problem 2.*

*Proof.* The number  $\tilde{n}$  is composite, because  $b \mid \tilde{n}$  and  $1 < b < \tilde{n}$ . For the same reason it will remain composite whenever we decide to leave the least significant digit intact. To complete the proof, observe that altering two digits of  $\tilde{n}$ , one of which is the least significant one, leads us to a number that is obtained by exactly the same alteration of  $n$ .  $\square$

Observe that, for  $n$  having 1 nonzero digit, either its least significant digit is zero (then  $\tilde{n} = n$  and Lemma 3 holds trivially), or nonzero (then  $n$  can be transformed into 1, which is not composite, and Lemma 3 holds trivially).

Additionally, for  $n$  having 4 nonzero digits, the only situation when Lemma 3 fails is when the second least significant digit of  $n$  is 1.

This means that, when looking for a solution to Problem 2, it is sufficient to consider a relatively small number of cases where  $n$  has 2, 3, or 4 nonzero digits (in the latter case with second least significant digit being 1) and the numbers divisible by  $b$ . The former can be done directly. The latter is accomplished by the following algorithm.

**Algorithm 4.** Given integers  $b \geq 2$ ,  $e \geq 0$ ,  $u \geq 0$ , to find all solutions  $n \in b\mathbb{Z} \cap [ub^{e+1}, (u+1)b^{e+1})$  to Problem 2:

1. For each integer  $k \in [0, b^{e+1})$ :
2.   If  $ub^{e+1} + k$  is prime: set  $a_k \leftarrow 1$ ; otherwise set  $a_k \leftarrow 0$ .
3. For each integer  $k \in [0, b^e)$ :
4.   If any of  $a_{bk}, a_{bk+1}, \dots, a_{bk+b-1}$  is 1: set  $b_k \leftarrow 1$ ; otherwise set  $b_k \leftarrow 0$ .
5. For each integer  $k \in [0, b^e)$ :
6.   Set  $c_k \leftarrow 1$ .
7. For each integer  $\ell \in [0, b^e)$  with  $k$  differing from  $\ell$  at at most one digit:
8.   If  $b_\ell = 1$ : set  $c_k \leftarrow 0$ .
9. For each integer  $k \in [0, b^e)$  with  $c_k = 1$ :
10. Check directly if  $b(ub^e + k)$  is a solution and output if it is.

The steps 1–2 require finding all prime numbers in an interval. We have used the following algorithms, all with running time close to  $O(b^e)$ :

- (a) a straightforward implementation of the sieve of Eratosthenes,
- (b) the implementation of the sieve of Atkin from [Ber 2007],

(c) our own implementations of the sieve of Atkin with  $W = 12$ ,  $W = 60$ , and  $W = 420$  (see [Atk-Ber 2004] for details).

The author of [Ber 2007] claims that the program works for primes up to  $10^{15}$ . The code is pretty complicated, so we could not figure out whether it works past that boundary. That is why we decided to create our own implementation. Surprisingly, the version  $W = 12$  worked best for really large numbers.

The steps 7–8 can be done in  $O(eb)$  time. Consequently, we implemented the whole loop 6–8 in  $O(eb^e)$  time.

In practice, the step 10 is involved only for a couple values of  $k$ , so its influence on running time is negligible.

If we are interested in finding the smallest solution, it is enough to call Algorithm 4 for fixed  $b, e$  and sequential units  $u$ . However, due to limited computational resources, it is more important to find any solution rather than to prove that it is minimal. Therefore, we first scanned the units, for which the probability of finding a solution is high.

The method we used is neither strict nor formal, but worked in practice. For the sake of the estimation, we assumed that primality of numbers is a result of a sequence of independent random experiments. The probability of a number in unit  $u$  being prime is  $(\pi((u+1)b^{e+1}) - \pi(ub^{e+1}))b^{-(e+1)}$ . Using the approximation  $\pi(n) \approx n/\ln(n)$  we obtained the approximate value of the probability  $p(u)$  of a number from block  $u$  being a solution to Problem 2. Then, we considered the blocks in order of decreasing  $p(u)$ . The computation done is summed up in Table 4.

$b$	$e$	units scanned	solution unit
2	6	0	0
3	6	0	0
4	7	0	0
5	9	0	0
6	10	0	0
7	10	0–39	39
8	10	0–218	218
9	10	0–1542	1542
10	9	0–97773183	97773183

TABLE 4. Summary of units considered

## REFERENCES

- [Atk-Ber 2004] A.O.L. Atkin, D.J. Bernstein, *Prime sieves using binary quadratic forms*, Math. Comp. 73 (2004), 1023–1030.
- [Ber 2007] D.J. Bernstein, <http://cr.yp.to/primegen.html> (seen on June 19th, 2007).
- [Erd 1952] P. Erdős, *On a problem concerning congruence systems* (Hungarian), Mat. Lapok 3 (1952), 122–128.
- [Mat 1977] Problem corner (Polish), Matematyka 1(147) (1977), p. 55.
- [Mat 1978] Problem corner (Polish), Matematyka 6(158) (1978), p. 362.
- [Mat 1980] Problem corner (Polish), Matematyka 6(170) (1980), p. 369.
- [Sch 1992] A. Schinzel, *On a problem by Wacław Sierpiński* (Polish), Gradient 1 (1992), No 2, 6–9.

INSTITUTE OF MATHEMATICS, JAGIELLONIAN UNIVERSITY, REYMONTA 4, 30–150 KRAKÓW,  
POLAND

*E-mail address:* Witold.Jarnicki@im.uj.edu.pl

*Current address:* Google R&E, Kraków, Krupnicza 16, 31–123 Kraków

*E-mail address:* witoldjarnicki@google.com

THEORETICAL COMPUTER SCIENCE DEPARTMENT, JAGIELLONIAN UNIVERSITY, GRONOSTAJOWA  
3, 30–387 KRAKÓW, POLAND

*E-mail address:* maze@tcs.uj.edu.pl

*Current address:* Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043

*E-mail address:* maze@google.com